



BLOCKCHAINLAB

Programmiamo la Blockchain

Strumenti e tecniche per
programmare sulla blockchain

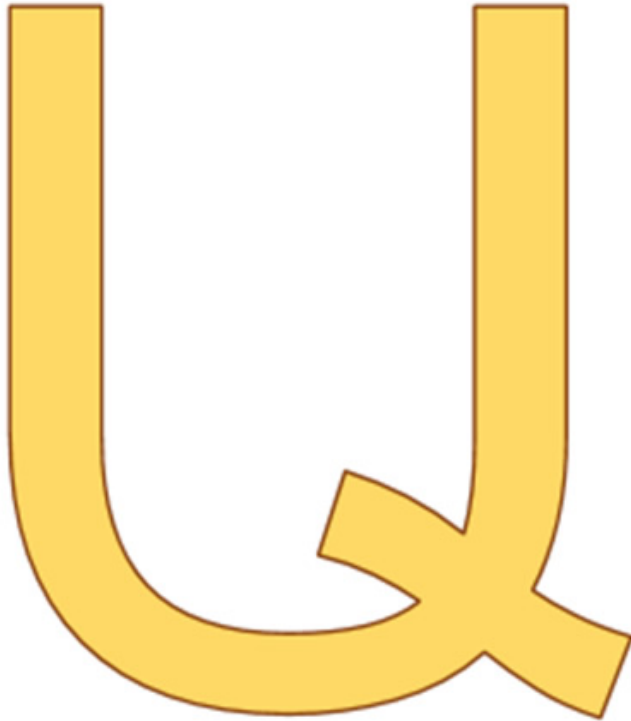
Francesco La Regina – Enzo Pastorelli
BLOCKCHAINLAB

Creazione di una blockchain privata

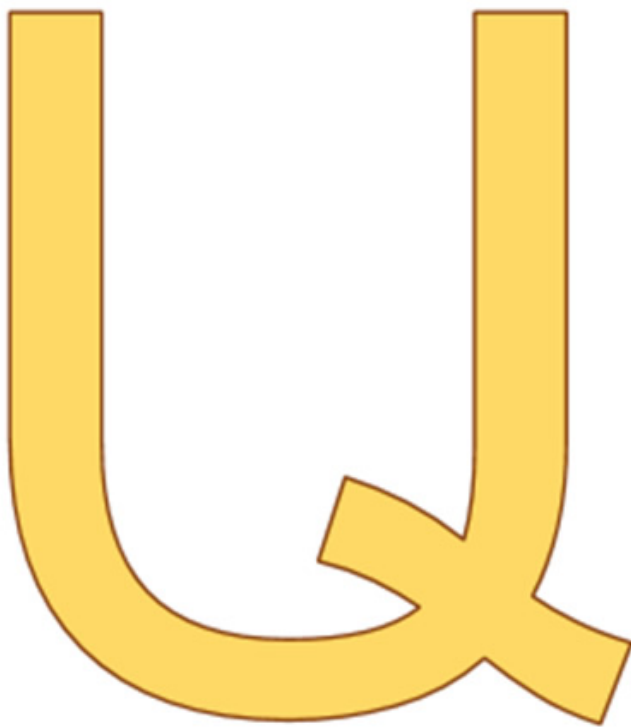
Ethereum è una piattaforma decentralizzata che gestisce contratti intelligenti: applicazioni che si eseguono esattamente come se fossero programmate senza possibilità di interruzioni, censura, frode o interferenze di terzi.

Go Ethereum (o [geth](#)) è una delle tre implementazioni originali (insieme a C ++ e Python) del protocollo Ethereum. È scritto in Go, completamente open source e licenziato sotto la GNU LGPL v3. Go Ethereum è disponibile come un client standalone chiamato geth che è possibile installare praticamente su qualsiasi sistema operativo o come una libreria che può essere incorporata nei progetti Go, Android o iOS.

<https://geth.ethereum.org/downloads/>



<https://www.unicalcoin.it/>



<https://www.unicalcoin.it/istruzioni/>

Installare geth su os x

Per installare geth su Mac OS X, utilizziamo https://brew.sh/index_it. Homebrew installa le cose di cui di solito si ha bisogno che Apple non ha fatto (un gestore di pacchetti tipo apt su Debian GNU/Linux).

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
brew tap ethereum/ethereum
```

```
brew install ethereum
```

Per GNU/Linux o Windows vi rimando alle istruzioni di installazione ufficiali:

<https://ethereum.github.io/go-ethereum/install/>

Installare geth su os x

```
mkdir /Users/xxxxxx/unicalcoin/
```

```
mkdir /Users/xxxxxx/unicalcoin/chain
```

Creazione del primo blocco (genesis block)

Una blockchain di ethereum differisce da un'altra in base al blocco di genesi. Una blockchain inizia con un blocco di genesi e continua a crescere: ogni blocco si riferisce a quello precedente. Nella blockchain pubblica di ethereum, il blocco di genesi è stato creato il 20 luglio 2015. Per costruire la nostra blockchain, dobbiamo creare il nostro blocco di genesi.

All'interno cartella unicalcoin andremo a creare un file vuoto che chiameremo genesis.json

Blocco di genesi

[illegible]

Blocco di genesi

chainid: questo fornisce un modo per inviare transazioni che lavorano su ethereum senza lavorare su ETC (ethereum classic) o sul testnet Morden. EIP 155 suggerisce di seguire i valori catena per reti diverse: ethereum mainnet (1), morden /expanse mainnet (2), ropsten (3), rinkeby (4), rootstock mainnet (30), rootstock testnet (31), kovan (42), ethereum classic mainnet (61), ethereum classic testnet (62), geth canali privati (1337 by default). nel nostro caso ho usato 1999, per evitare problemi di compatibilità con altri strumenti che vedremo più avanti.

homesteadBlock: il valore 0 indica che utilizza la release ethereum homestead. Homestead è la seconda generazione - il 16 ottobre 2017, l'ethereum ha fatto il passaggio al byzantium.

eip155Block: il valore 0 indica che questo blockchain supporta l'EIP (proposta di miglioramento dell'etereum) 155. IIP descrive gli standard per la piattaforma ethereum, tra cui le specifiche del protocollo di base, le API del client e gli standard.

eip158Block: idem, il valore 0 dice che questa blockchain supporta l'EIP 158.

difficulty: un valore corrispondente al livello di difficoltà applicato durante la scoperta di questo blocco.

gasLimit: il gas è il prezzo per eseguire un'operazione o un contratto in Ethereum. Ogni istruzione inviata alla Ethereum Virtual Machine (EVM) per elaborare una transazione o un contratto intelligente costa una quantità specifica di gas. Fino a qualche mese fa il prezzo di 1 gas era fissato a 10 Szabo, ovvero 1 / 100.000 di un Ether. Se la quantità richiesta di gas non viene fornita all'operazione, esso non viene eseguita. Quando si effettua una transazione, si specifica un limite di gas - cioè il gas massimo che tutte le operazioni corrispondenti a tale transazione possono consumare. Il parametro gasLimit nel blocco specifica, il gasLimit aggregato da tutte le transazioni incluse nel blocco.

alloc: questo consente di preallocare un certo numero di ether a uno o più conti. Nel blocco di genesi di cui sopra, la preallocazione viene fatta al conto che abbiamo creato all'inizio (attenzione - da sostituire con il vostro indirizzo).

Inizializzare la rete

```
geth --datadir=/Users/XXX/unicalcoin/chain init /Users/XXX/unicalcoin/genesis.json
```

```
geth --datadir=/Users/XXX/unicalcoin/chain account new
```

WARN [10-22|10:19:39] No etherbase set and no accounts found as default Your new account is locked with a password. Please give a password. Do not forget this password.

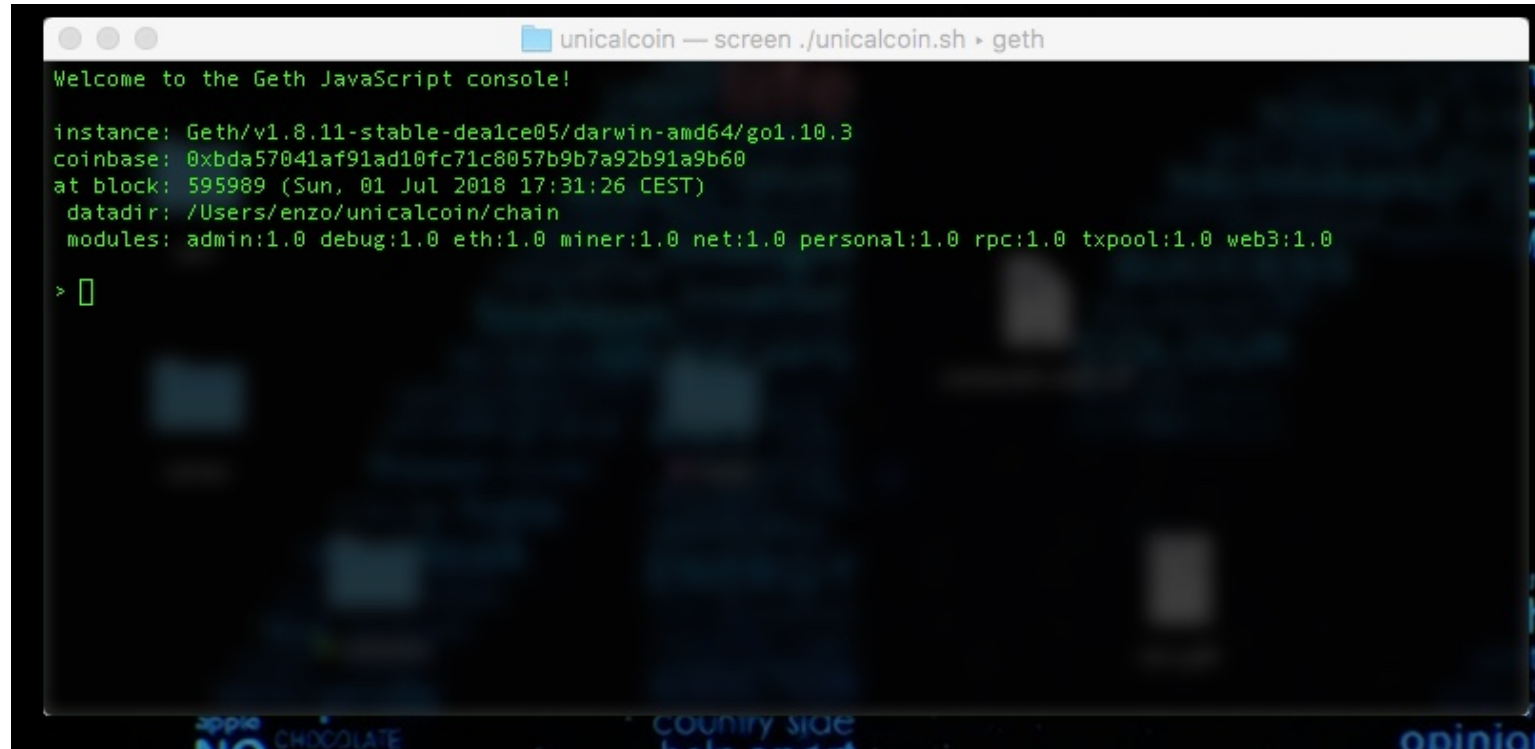
Passphrase:

Repeat passphrase:

Address: {cc92817ad1892cd8c98ea84394ce401ec93ac1a2}

Console geth

```
geth --datadir=/Users/xxx/unicalcoin/chain --bootnodes  
"enode://98d0eefc97666ddb297f1a6c80f02f7e7c2312b8a24cd045440d18d6277cf0de1be0ca22551bb785ff9549c0a90  
685c47647f2587e239a76a0f1f8ff9e8e5ed9@160.97.62.236:30302" console 2>>/Users/xxx/unicalcoin/log.log --port  
30303 --rpc --rpcport 8584 --rpcaddr 0.0.0.0 --rpcorsdomain "*" --networkid 9654321 --ipcpath  
/Users/xxx/Library/Ethereum/geth.ipc
```

A screenshot of a terminal window titled "unicalcoin — screen ./unicalcoin.sh ▶ geth". The terminal displays the Geth JavaScript console output. The text is as follows:
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.11-stable-dealce05/darwin-amd64/go1.10.3
coinbase: 0xbda57041af91ad10fc71c8057b9b7a92b91a9b60
at block: 595989 (Sun, 01 Jul 2018 17:31:26 CEST)
datadir: /Users/enzo/unicalcoin/chain
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>

Console geth – Comandi base

```
> eth.accounts
["0xbda57041af91ad10fc71c8057b9b7a92b91a9b60"]
```

```
> eth.getBalance("0xbda57041af91ad10fc71c8057b9b7a92b91a9b60")
1000000000000000000000000
```

```
> personal.unlockAccount("xbda57041af91ad10fc71c8057b9b7a92b91a9b60", "password")
```

```
> eth.sendTransaction({from:eth.accounts[0], to:"0x629b0ce35a2a1354c9f59d7164a20832b173dc59",  
value: web3.toWei(1, "ether")})
```

```
> miner.start(2)
```

```
>miner.stop ()
```

Metamask

<https://metamask.io/>

MetaMask è un portafoglio di ethereum, che si può utilizzare come estensione di Chrome Firefox. Per connettersi al nodo ethereum, MetaMask utilizza la libreria web3.js che offre un modo per connettersi all'interfaccia RPC.

Scarichiamo e attiviamo l'estensione Metamask su Google Chrome;

Accettiamo quanto scritto, impostiamo la password e colleghiamoci alla rete pubblica;

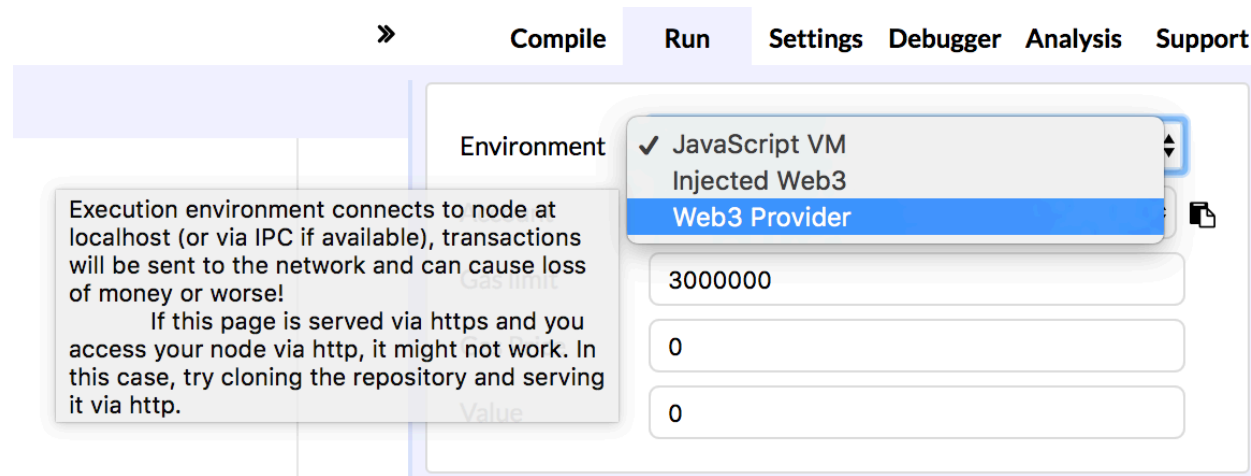
Colleghiamoci alla rete privata.

<http://160.97.62.236:8584>

Scrivere uno smart contract

<https://remix.ethereum.org/>

Ora possiamo scrivere il nostro primo smart contract da eseguire sulla nostra rete ethereum privata. Per farlo useremo Solidity, il linguaggio di programmazione più diffuso per scrivere contratti intelligenti ethereum.



Scrivere uno smart contract

<https://remix.ethereum.org/>

```
pragma solidity ^0.4.18;

contract cometichiemi {

    string Nome;
    uint eta;

    event dati(
        string nome,
        uint eta
    );

    function setDati(string _Nome, uint _eta) public {
        Nome = _Nome;
        eta = _eta;
        emit dati(_Nome, _eta);
    }

    function getDati() view public returns (string, uint) {
        return (Nome, eta);
    }
}
```

```
pragma solidity ^0.4.18;

contract ListaPersone {

    event dati(
        uint personeid,
        string nome,
        uint eta
    );

    struct Persona {
        string Nome;
        uint eta;
    }

    Persona[] public persone;

    mapping (uint => address) public personeToOwner;
    mapping (address => uint) ownerPersoneCount;

    function createPersone(string _Nome, uint _eta) public {
        require(ownerPersoneCount[msg.sender] == 0);
        uint id = persone.push(Persona(_Nome, _eta)) - 1;
        personeToOwner[id] = msg.sender;
        ownerPersoneCount[msg.sender]++;
        emit dati(id, _Nome, _eta);
    }
}
```



<https://www.inzhoop.com/>

Unicalcoin sul tuo smartphone

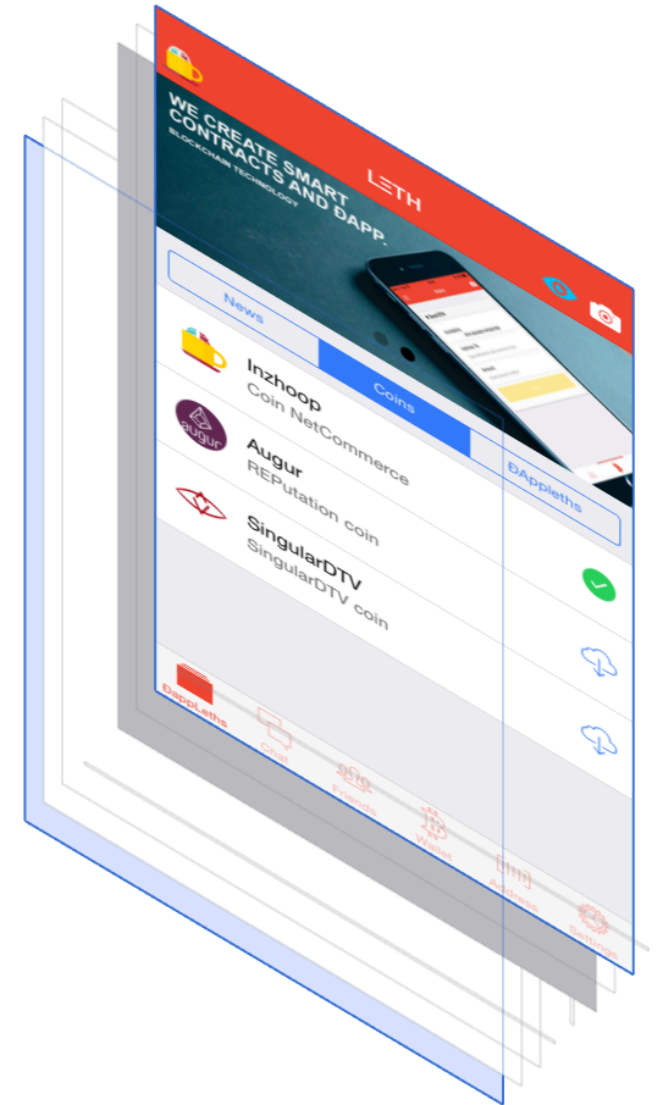
Scarica il wallet LETH da [inzhoop.com](https://www.inzhoop.com/)

Avvia l'app e, la prima volta, crea un nuovo account

Vai in Impostazioni e seleziona l'Indirizzo Provider Host
(sotto Tipo di Host)

Aggiungi Custom Host Address:

<http://160.97.62.236:8584>



A stage with red curtains and a spotlight on the floor. The word "Grazie!" is written in green text in the center of the stage.

Grazie!



BLOCKCHAINLAB